

(19)



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: **06309265 A**

(43) Date of publication of application: **04.11.94**

(51) Int. Cl.  
**G06F 13/10**  
**G06F 3/00**  
**G06F 9/06**

(21) Application number: **05101062**

(71) Applicant: **SANYO ELECTRIC CO LTD**

(22) Date of filing: **27.04.93**

(72) Inventor: **TOMIZAWA SUGIO**

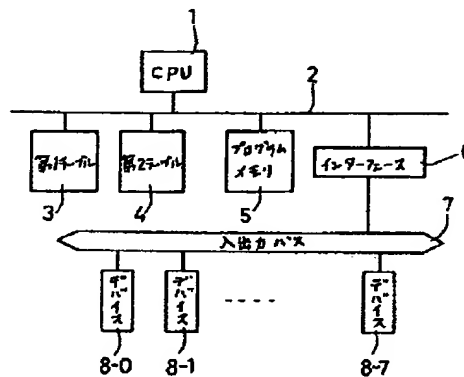
**(54) DYNAMIC SELECTION METHOD FOR DRIVER  
PROGRAM OF INFORMATION PROCESSOR**

**(57) Abstract:**

**PURPOSE:** To dynamically decide the correspondence between an input/output device and the driver program which controls the input/output device.

**CONSTITUTION:** An input/output request is produced from an application program to a device having the ID number with which a common driver program is registered in a 1st table 3. Thus the common driver program that received the control right from an OS produces an inquiry command to the device having the preceding ID number in order to acquire the device information. Based on this information, the corresponding driver program is decided by making reference to a 2nd table 4. Then the entry address of the decided program is registered in the table 3 in response to the relevant ID number. At the same time, the driver program jumps to the registered address.

COPYRIGHT: (C)1994,JPO



(19) 日本国特許庁 ( J P )

(12) 公開特許公報 ( A )

(11) 特許出願公開番号

特開平6-309265

(43) 公開日 平成 6 年 (1994) 11 月 4 日

(51) Int. Cl. <sup>5</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G06F 13/10	330	B 8133-5B		
3/00		A 7165-5B		
9/06	410	S 9367-5B		

審査請求 未請求 請求項の数 3 O L (全 6 頁)

(21) 出願番号	特願平5-101062	(71) 出願人	000001889 三洋電機株式会社 大阪府守口市京阪本通 2 丁目 5 番 5 号
(22) 出願日	平成 5 年 (1993) 4 月 27 日	(72) 発明者	富沢 杉雄 大阪府守口市京阪本通 2 丁目 18 番地 三洋 電機株式会社内
		(74) 代理人	弁理士 西野 卓嗣

(54) 【発明の名称】 情報処理装置におけるドライバプログラムの動的選択方法

(57) 【要約】

【目的】 入出力デバイスとそのデバイスを制御するドライバプログラムとの対応を動的に決定する。

【構成】 第 1 テーブル 3 において共通ドライバプログラムを登録した I D 番号のデバイスに対して、アプリケーションプログラムによって入出力要求が発生すると、OS から制御を渡された共通ドライバプログラムは、当該 I D 番号のデバイスに対してインクエリコマンドを発行することによってデバイス情報を取得する。そして、その情報に基づいて第 2 テーブル 4 を参照して対応するドライバプログラムを決定すると、そのドライバプログラムのエントリアドレスを第 1 テーブル 3 において当該 I D 番号に対応させて登録するとともに、そのアドレスにジャンプする。

## 【特許請求の範囲】

【請求項 1】 特定コマンドに応答して固有のデバイス情報を返すことのできる複数の入出力デバイスの少なくとも 1 つを接続すると共に、前記複数の入出力デバイスの各々に対応する複数のドライバプログラムと、前記入出力デバイスを識別する為の識別番号と前記ドライバプログラムとの対応を表す第 1 のテーブルとを備え、前記第 1 のテーブルを参照して前記識別番号に対応するドライバプログラムによって前記入出力デバイスを制御する情報処理装置において、前記デバイス情報と前記ドライバプログラムとの対応を表す第 2 のテーブルと、共通ドライバプログラムとを設け、前記第 1 のテーブルに前記共通ドライバプログラムを登録しておき、該共通ドライバプログラムが登録された前記識別番号が入力されたとき、前記共通ドライバプログラムは、当該識別番号に対応する前記入出力デバイスに特定コマンドを発行し、取得した前記デバイス情報を用いて前記第 2 のテーブルを参照して対応する前記ドライバプログラムを決定し、該ドライバプログラムを前記第 1 のテーブルにおいて当該識別番号に対応する共通ドライバプログラムに代えて登録すると共に、該ドライバプログラムによって前記入出力デバイスを制御することを特徴とする情報処理装置におけるドライバプログラムの動的選択方法。

【請求項 2】 前記第 1 のテーブルは、対応する前記入出力デバイスが置き換えられる可能性がある前記識別番号について、前記共通ドライバプログラムを登録することを特徴とする請求項 1 記載の情報処理装置におけるドライバプログラムの動的選択方法。

【請求項 3】 前記デバイス情報は、製造メーカー名及び製品名であることを特徴とする請求項 1 または 2 記載の情報処理装置におけるドライバプログラムの動的選択方法。

## 【発明の詳細な説明】

## 【 0 0 0 1 】

【産業上の利用分野】 本発明は、接続した入出力デバイスのそれぞれを、各デバイス毎に対応するドライバプログラムで制御する情報処理装置に係り、特に、対応するドライバプログラムを動的に選択する方法に関する。

## 【 0 0 0 2 】

【従来の技術】 情報処理装置において、例えば SCSI バスに代表されるような標準入出力バスを介してそれぞれ異なる ID 番号（識別番号）を持つ各種入出力デバイス（以下、デバイスと略称する）を接続したときに、ドライバプログラムが各々のデバイスのデバイス情報を各 ID 番号毎に認識し、そのデバイス情報に従って各デバイスをアクセスすることが行われる。

【 0 0 0 3 】 このような方法を行うものとして、例えば、特開平 5 - 2 5 5 0 号公報に記載されたものがある。これは、接続されたデバイスのデータ転送ブロック長などのデバイス情報を認識するための特定コマンドを

発行する手段と、このコマンドを受けたデバイスからデバイス情報を通知する手段とを設けて、電源投入時に入出力バスに接続されている各デバイス毎に特定コマンドを発行し、それぞれのデバイス情報を取得しておいて、以降は、このデバイス情報に従って当該デバイスをアクセスするものである。

## 【 0 0 0 4 】

【発明が解決しようとする課題】 ハードディスク装置や、ストリーマテープ装置といったデバイス種類が同一であっても、製造メーカーが異なれば、例えば同じストリーマテープ装置でも転送ブロック長などが完全には一致しないのが実情である。また、オプションでサポートするコマンドや、エラー発生時のエラー情報の内容やその対処方法なども違うために、従来例のように 1 つのドライバプログラムによって様々な製造メーカーの装置に対応し、デバイスそれぞれの性能を最大限に引き出すようにするには、ドライバプログラムが大きく複雑になる問題があった。

【 0 0 0 5 】 そうかと言って、接続するデバイスそれぞれに専用のドライバプログラムを用意する単純な方法では、ある ID 番号のデバイスを別のデバイスに置き換えた場合には、その ID 番号のデバイスを制御するドライバプログラムも、同時に新しいデバイス専用のものに置き換えることが行われるが、この場合、オペレーティングシステムに対するドライバプログラムの登録を変更して、オペレーティングシステムをジェネレーションし直す必要があり、多くの手間と時間を要する欠点があった。

【 0 0 0 6 】 そこで本発明は、入出力バスに接続したデバイスを他のデバイスに置き換えた場合でも、例えばシステムの起動後に初めてそのデバイスに対する入出力要求が発生した時に、当該デバイスに対応するドライバプログラムを決定するようにして、デバイスの変更に柔軟に対応できるシステムを実現できるようにするものである。

## 【 0 0 0 7 】

【課題を解決するための手段】 本発明は、特定コマンドに응答して固有のデバイス情報を返すことのできる複数の入出力デバイスの少なくとも 1 つを接続すると共に、前記複数の入出力デバイスの各々に対応する複数のドライバプログラムと、前記入出力デバイスを識別する為の識別番号と前記ドライバプログラムとの対応を表す第 1 のテーブルとを備え、前記第 1 のテーブルを参照して前記識別番号に対応するドライバプログラムによって前記入出力デバイスを制御する情報処理装置において、前記デバイス情報と前記ドライバプログラムとの対応を表す第 2 のテーブルと、共通ドライバプログラムとを設け、前記第 1 のテーブルにおいて前記共通ドライバプログラムを登録しておき、該共通ドライバプログラムが登録された前記識別番号が入力されたとき、前記共通ドライバ

プログラムは、当該識別番号に対応する前記入出力デバイスに特定コマンドを発行し、取得した前記デバイス情報を用いて前記第2のテーブルを参照して対応する前記ドライバプログラムを決定し、該ドライバプログラムを前記第1のテーブルにおいて当該識別番号に対応する共通ドライバプログラムに代えて登録すると共に、該ドライバプログラムによって前記入出力デバイスを制御するようにして、上記課題を解決するものである。

【0008】また、前記第1のテーブルは、対応する前記入出力デバイスが置き換えられる可能性がある前記識別番号について、前記共通ドライバを登録するようにし、そして、前記デバイス情報は、製造メーカ及び製品名を前記第2のテーブルに登録することで、上記課題を解決するようにしている。

【0009】

【作用】本発明では、第1のテーブルにおいて共通ドライバプログラムが登録された識別番号に対する入出力要求があった場合、共通ドライバプログラムは当該識別番号の入出力デバイスに特定コマンドを発行し、そのデバイスから取得したデバイス情報に基づいて第2のテーブルを参照して対応するドライバプログラムを決定する。そして、そのドライバプログラムを第1のテーブルにおいて当該識別番号に対応する共通ドライバプログラムに代えて登録すると共に、そのドライバプログラムによってデバイスを制御する。

【0010】

【実施例】図4は本実施例の情報処理装置の構成を示すブロック図である。図において1は各種処理を実行するCPU、2はシステムバス、5は各種プログラムを記憶するプログラムメモリ、6はインターフェース、7はSCSIバスなどの入出力バスであり、この入出力バス7にはそれぞれID番号によって識別される複数のデバイス8-0~8-7が接続されている。本実施例では、入出力バス7には8台までデバイスを接続することができ、それぞれID番号としてID#0~ID#7が割当てられている。これらのデバイスは、インクエリコマンド(inquiryコマンド)が入力されると、それに応答して、自己の固有のデバイス情報を返すよう構成されており、デバイス情報としては、例えば図6に示す情報を用いている。ここで、デバイスタイプは、そのデバイスがハードディスクドライブなどのランダムアクセス可能なものか、ストリーマテープ装置のような順次アクセスするものかを区別するものである。規格バージョンはSCSI規格などのそのデバイスがサポートするインターフェース規格のバージョンを表すものである。メーカ情報はそのデバイスの製造メーカ名であり、プロダクト情報はそのデバイスの製品名(商品名)を記述したものである。

【0011】また、本実施例では、デバイス8-0~8-7各々に対応して、それぞれ個別にデバイスを制御す

る複数の専用のドライバプログラムを用意し、これらの専用ドライバプログラムと、全デバイスに共通の共通ドライバプログラムとを前述のプログラムメモリ5に記憶している。

【0012】そのプログラムメモリ5の内容を示したのが図5である。アドレス10000未満にオペレーティングシステム(OS)を記憶し、アドレス10000以降に共通ドライバプログラム(CDP)と、各デバイスに対応するドライバプログラム(DP0~DP9)を記憶している。そして、アドレス22900以降にはアプリケーションプログラムが記憶されている。

【0013】更に、図4において、3はID番号とドライバプログラムの対応関係を表す第1テーブル、4はデバイス情報とドライバプログラムの対応関係を表す第2テーブルである。

【0014】図2は、第1テーブル3の内容を示しており、ID番号とそれに対応するドライバプログラムのエントリアドレスの対を、ID番号の昇順に並べた構成である。図3(a)は、本実施例の情報処理装置を起動した直後の第1テーブル3の内容である。あらかじめデバイスの変更が予想されるID#0に対応して、共通ドライバプログラムのエントリアドレス10000が登録してある。

【0015】そして、図3は第2テーブル4の内容を示し、接続される可能性があるデバイスについて、前述のデバイス情報のうちメーカ及びプロダクト情報と、そのデバイス専用のドライバプログラムのエントリアドレスとの対応を表すものである。

【0016】本実施例では、ID#0のデバイスとして、A社のa製品またはB社のb製品のいずれかが接続されるものとする。

【0017】以下、図1のフローチャートを参照しながら本実施例の動作を説明する。

【0018】システムの起動後、アプリケーションプログラムによってID#0に対する入出力要求が発生したとすると、オペレーティングシステムは、第1テーブル3を参照し、ID#0に対応して登録されているエントリアドレス10000、即ち共通ドライバプログラムに制御を移行する。共通ドライバプログラムは、図1のフローチャートに従う次のような処理を実行する。

【0019】まず、ID#i(i=0)のデバイス(ここではデバイス8-0)にインターフェース6及び入出力バス7を介してインクエリコマンドを発行し、コマンドを受け取ったデバイス8-0が自身のデバイス情報を返すことで、共通ドライバプログラムはデバイス情報を取得する(ステップS1)。そして、取得したデバイス情報のうち、メーカ及びプロダクト情報に基づいて第2テーブル4を参照し、対応するドライバプログラムのエントリアドレスを取得する(ステップS3)。ここで、デバイス8-0がA社のa製品であるとする、ステッ

ブS3においてエントリアドレス20500が得られることになる。

【0020】次に共通ドライバプログラムは、第1テーブル3におけるID#i（ここではi=0）に対応するエントリアドレスとして、自身のエントリアドレス10000に代えて、ステップS3において取得したエントリアドレス20500を登録し（ステップS5）、しかる後にそのエントリアドレスにジャンプする（ステップS6）。即ち、ID番号ID#0が割当てられたデバイス8-0専用のドライバプログラムのエントリアドレスを、第1テーブル3においてID#0に対応させて登録しておいて、そのドライバプログラムに制御を移行するのである。こうして新しい内容に書き替えられた第1テーブル3を示すのが図2（b）である。

【0021】従って、これ以降にID#0に対する入出力要求が発生した場合には、第1テーブル3の内容は既書き替えられているから、オペレーティングシステムからは直接にデバイス8-0専用のドライバプログラム（この場合DP8）に制御が移行することになる。

【0022】もしここで、ID#0のデバイスとしてB社のb製品が接続されていたならば、ステップS3においてエントリアドレス21700が得られるから、次にステップS5において、共通ドライバプログラムは、第1テーブルのID#0に対応してエントリアドレス21700を登録し、そして、そのエントリアドレスにジャンプする（ステップS6）。この結果、これ以降は、ID#0のデバイスのドライバプログラムは、図5に示すアドレス21700に記憶したDP9が使われるのである。

【0023】なお、図1のステップS1において、ID#iのデバイスが存在しないなどのエラーが発生した場合は、ステップS2からステップS7に進み、また、ステップS3において該当するメーカ及びプロダクトが見つからなかった場合は、同様にステップS4からS7に進み、共通ドライバプログラムはオペレーティングシステムにエラーを返す。

【図2】

(a)		(b)	
ID番号	エントリアドレス	ID番号	エントリアドレス
ID#0	10000	ID#0	20500
ID#1	12100	ID#1	12100
ID#2	13200	ID#2	12300
⋮	⋮	⋮	⋮
ID#7	19300	ID#7	19300

【0024】ところで、本実施例では、あるID番号に対する入出力要求が発生した時点で、そのID番号に対応するドライバプログラムを共通ドライバプログラムによって決定しているが、これをシステムの起動時に各ID番号に対して順にダミーの入出力要求を発行することで、あらかじめID番号とドライバプログラムの対応を決定しておくようにもできる。

## 【0025】

【発明の効果】本発明によれば、入出力要求のあったデバイスのデバイス情報を取得し、そのデバイス情報に基づいて対応するドライバプログラムを選択するようにしたから、入出力バスに接続する入出力デバイスの変更に柔軟に対応できる。そして、その結果、システム構成の自由度が著しく向上する。

【0026】また、各々の入出力デバイスに専用のドライバプログラムを使用することで、個々のデバイスの性能をフルに発揮させることができ、もってシステム全体の性能が向上する。

## 【図面の簡単な説明】

【図1】実施例における共通ドライバプログラムの処理手順を示すフローチャートである。

【図2】第1テーブルの一例を示す図である。

【図3】第2テーブルの一例を示す図である。

【図4】実施例の情報処理装置の構成を示すブロック図である。

【図5】実施例におけるプログラムメモリ構成を示す図である。

【図6】デバイス情報の一例を示す図である。

## 【符号の説明】

- 1 CPU
- 3 第1テーブル
- 4 第2テーブル
- 5 プログラムメモリ
- 7 入出力バス
- 8-0～8-7 デバイス

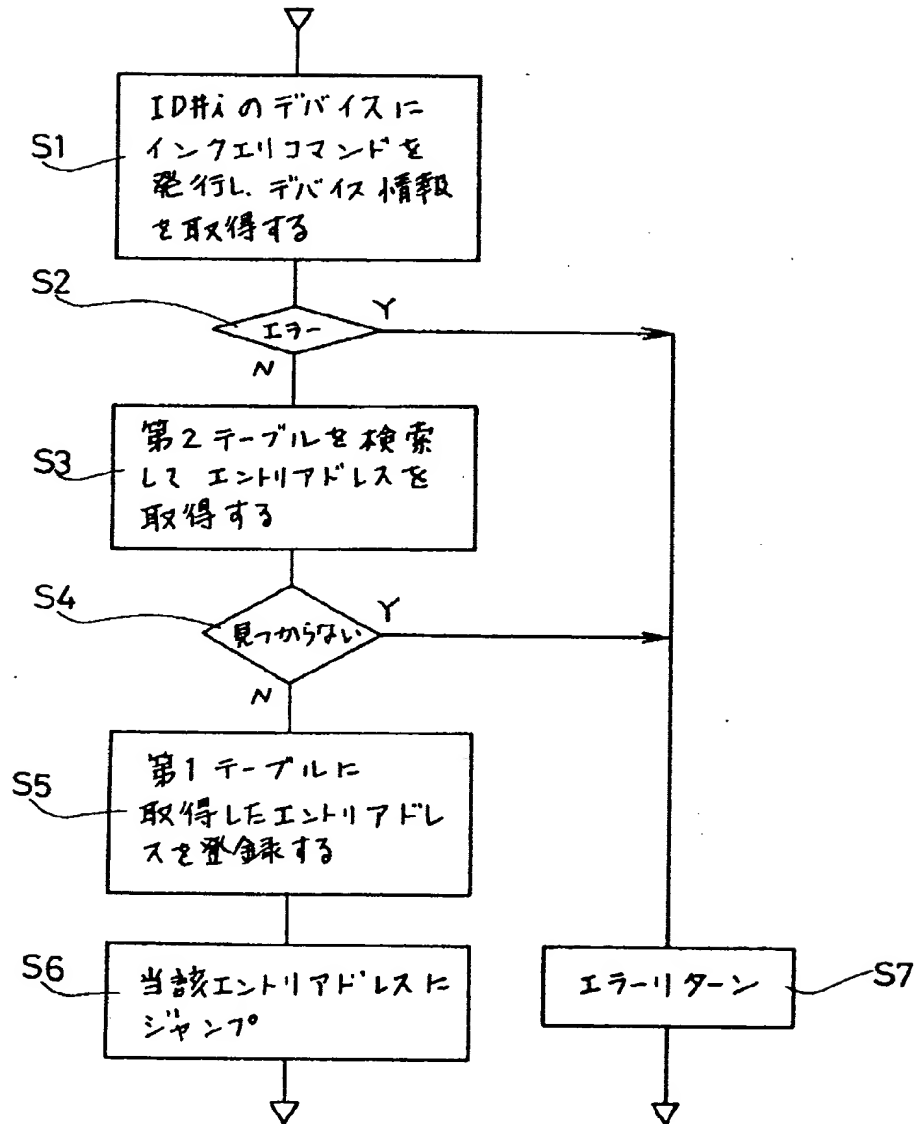
【図3】

メーカ	プログラム	エントリアドレス
A社	A	20500
B社	b	21700
⋮	⋮	⋮

【図5】

アドレス	内容
0	DS
10000	GDP
10900	DP0
12100	DP1
13300	DP2
⋮	⋮
19300	DP7
20500	DP8
21700	DP9
22900	DP10
⋮	⋮

【図1】



【図6】

デバイスタイプ
親塔バージョン
メーカー情報
プロダクト情報
Xの他

【図 4】

